LOAN DOCUMENT

PHOTOGRAPH THIS SHEET

LEVEL

INVENTORY

DTIC ACCESSION NUMBER

DOCUMENT IDENTIFICATION

DISTRIBUTION STATEMENT

ACCESSION FOR
NTIS GRA&I
DTIC TRAC
UNANNOUNCED
JUSTIFICATION

BY
DISTRIBUTION/
AVAILABILITY CODES

DISTRIBUTION	AVAILABILITY AND/OR SPECIAL	
A-1		

DISTRIBUTION STAMP

DATE ACCESSIONED

H A N D L E W I T H C A R E

Reproduced From
Best Available Copy

DATE RETURNED

19981223 038

DATE RECEIVED IN DTIC

REGISTERED OR CERTIFIED NUMBER

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC

DTIC FORM JUN 90 70A

DOCUMENT PROCESSING SHEET

PREVIOUS EDITIONS MAY BE USED UNTIL STOCK IS EXHAUSTED.

LOAN DOCUMENT

ESCAPE SYSTEM TRAJECTORY SIMULATION

FINAL REPORT

CDRL ITEM #A005

Task Order No. 7

Contract N62269-78-C-0191

Prepared for

NAVAL AIR DEVELOPMENT CENTER

Warminster, Pennsylvania

November 1979

CSC

COMPUTER SCIENCES CORPORATION

MATERIAL INSPECTION AND RECEIVING REPORT	1. PROC. INSTRUMENT IDEN (CONTRACT) N62269-78-C-0191	(ORDER) NO. 0007	6. INVOICE NO. N/A DATE	7. PAGE 1 OF 1 / 8. ACCEPTANCE POINT D

2. SHIPMENT NO.	3. DATE SHIPPED	4. B/L TCN	5. DISCOUNT TERMS
CSC/0007	31DEC79	9	N/A

9. PRIME CONTRACTOR CODE 051069	10. ADMINISTERED BY CODE S3910A
Defense Systems Center, Systems Division Computer Sciences Corporation 304 W. Route # 38, P.O. Box "N" Moorestown, NJ 08057	DCASD, Philadelphia P.O. Box 7699 Philadelphia, PA 19101
11. SHIPPED FROM (If other than 9) CODE FOB: D	12. PAYMENT WILL BE MADE BY CODE S3910A
SEE BLOCK 9	DCASR Office of Systems & Financial Management P.O. Box 7730 Philadelphia, PA 19101
13. SHIPPED TO CODE N62269	14. MARKED FOR CODE N62269
Receiving Officer Naval Air Development Center Warminster, Pennsylvania 18974	NADC/ 8131

15. ITEM NO	16. STOCK PART NO. DESCRIPTION (Indicate number of shipping containers - type of container - container number.)	17. QUANTITY SHIP/REC'D *	18. UNIT	19. UNIT PRICE	20. AMOUNT
A005	Final Report (Short Form)	3	ea		

21. PROCUREMENT QUALITY ASSURANCE

A. ORIGIN	B. DESTINATION	22. RECEIVER'S USE
☐ PQA ☐ ACCEPTANCE of listed items has been made by me or under my supervision and they conform to contract, except as noted herein or on supporting documents.	☒ PQA ☒ ACCEPTANCE of listed items has been made by me or under my supervision and they conform to contract, except as noted herein or on supporting documents.	Quantities shown in column 17 were received in apparent good condition except as noted.
DATE / SIGNATURE OF AUTH GOVT REP	DATE / SIGNATURE OF AUTH GOVT REP	DATE RECEIVED / SIGNATURE OF AUTH GOVT REP
TYPED NAME AND OFFICE	TYPED NAME AND TITLE	TYPED NAME AND OFFICE
		* If quantity received by the Government is the same as quantity shipped, indicate by [illegible] mark, if different, enter actual quantity received below quantity shipped and encircle.

23. CONTRACTOR USE ONLY

TASK 7011-0007

FINAL DELIVERY

ESCAPE SYSTEM TRAJECTORY SIMULATION

FINAL REPORT

CDRL ITEM #A005

Task Order No. 7

Contract N62269-78-C-0191

Prepared for

NAVAL AIR DEVELOPMENT CENTER

Warminster, Pennsylvania

November 1979

COMPUTER SCIENCES CORPORATION

101 Masons Mill Business Park

Huntingdon Valley, Pennsylvania 19006

BACKGROUND

Early in 1977, Computer Sciences Corporation (CSC) conducted a study for the Life Support Engineering Division (LSED) (603), Naval Air Development Center (NADC) under Contract N62269-75-0001, Task Order No. 47. The purpose of the study was to examine the feasibility of using a microprocessor based system to control the functions of a vertical seeking ejection system. In November, 1977, CSC issued a report containing the results of that study, titled "Microcomputer Controlled Ejection Seat Feasibility Study". CSC found the concept to be feasible and recommended further analysis. To this end, NADC initiated the purchase of the IMSAI PCS 80/30 microprocessor based software development system.

From January , 1978 to July, 1978, CSC, under Contract N62269-78-C-0191 Task Order No. 7, conducted a second effort for NADC and in July, issued a report entitled "Escape System Trajectory Simulation and Microprocessor Control System". This report contains the result of that effort, the purpose of which was to define the functional requirements of an ejection seat system, examine the reliability and maintainability aspects of the INTEL 8085 microprocessor for military use and outline a software demonstration program to perform the vertical seeking maneuver and to control the timing and sequencing of specific events during ejection. The data used to demonstrate the program was provided by the Naval Weapon Center, China Lake, CA.

In September 1978, CSC completed the development, testing, and demonstration of the program. In October, 1978, CSC issued an interim technical report entitled "Supplement to Escape System Trajectory Simulation and Microprocessor Control System", which documents the program that was developed and tested to demonstrate the capability of the INTEL 8085 microprocessor to control the functions of a vertical seeking ejection seat.

In June 1979, CSC was asked to implement the simulation of the vertical seeking ejection seat in the existing Aircrew Escape System Computer Simulation Program - ICARUS. In October 1979, CSC completed the implementation and preliminary testing of the revised version of the ICARUS Program.

1.0 INTRODUCTION

Three major functions were added to the ICARUS Program to give it the capability of simulating the vertical seeking ejection seat: a system which continuously measures the angular rotation of the seat/occupant; a procedure to update the seat/occupant position through a direction cosine update algorithm; a command decision function which determines the rocket roll and/or pitch commands by using the values in the direction cosine matrix and the angular rates of the seat/occupant system. The displacement of the rocket is required to get the seat/occupant system into an upright position within the shortest time possible.

The material presented herein assumes that the reader is familiar with the input file and basic structure of the ICARUS Program.

2.0 THE VERTICAL SEEKING MANEUVER

2.1 The Vertical Seeking Algorithm

There are 5 basic steps comprising the vertical seeking algorithm.

1. Initialize Seat/Occupant Orientation.
2. Integrate Rotational Rate Data.
3. Update Direction Cosines.
4. Generate Rocket Gimballing Commands.
5. Gimbal Rocket

2.2 Initialize the Orientation of the Seat/Occupant System

Two coordinate systems (shown in Figure 1) are involved in the computation of the vertical seeking ejection system, the Seat/Man Coordinate System (SMCS) and the Vertical Seeking Coordinate System (VSCS). There are three elements of the direction cosine matrix, D3(1), D3(2), and D3(3), which are defined at ejection initiation.

The direction cosine matrix (DVC) which transfers the Z-Axis of VSCS to the SMCS is given by

$$DVC = \begin{bmatrix} \mathrm{Cos}(C7) & 0 & \mathrm{Sin}(C7) \\ 0 & 1 & 0 \\ -\mathrm{Sin}(C7) & 0 & \mathrm{Cos}(C7) \end{bmatrix}$$

The direction cosine matrix ASEM which transfers a vector from the VSCS to the Earth Fixed Coordinate System (EFCS) is given by

$$[ASEM] = [DVC] \times [ASE]$$

SMCS - SEAT/MAN COORDINATE SYSTEM.

VSCS - VERTICAL SEEKING COORDINATE SYSTEM.

α_{31} = ANGLE (VERT, X_{VSCS})

α_{32} = ANGLE (VERT, Y_{VSCS})

α_{33} = ANGLE (VERT, Z_{VSCS})

D3(1) = COS (α31)

D3(2) = COS (α32)

D3(3) = COS (α33)

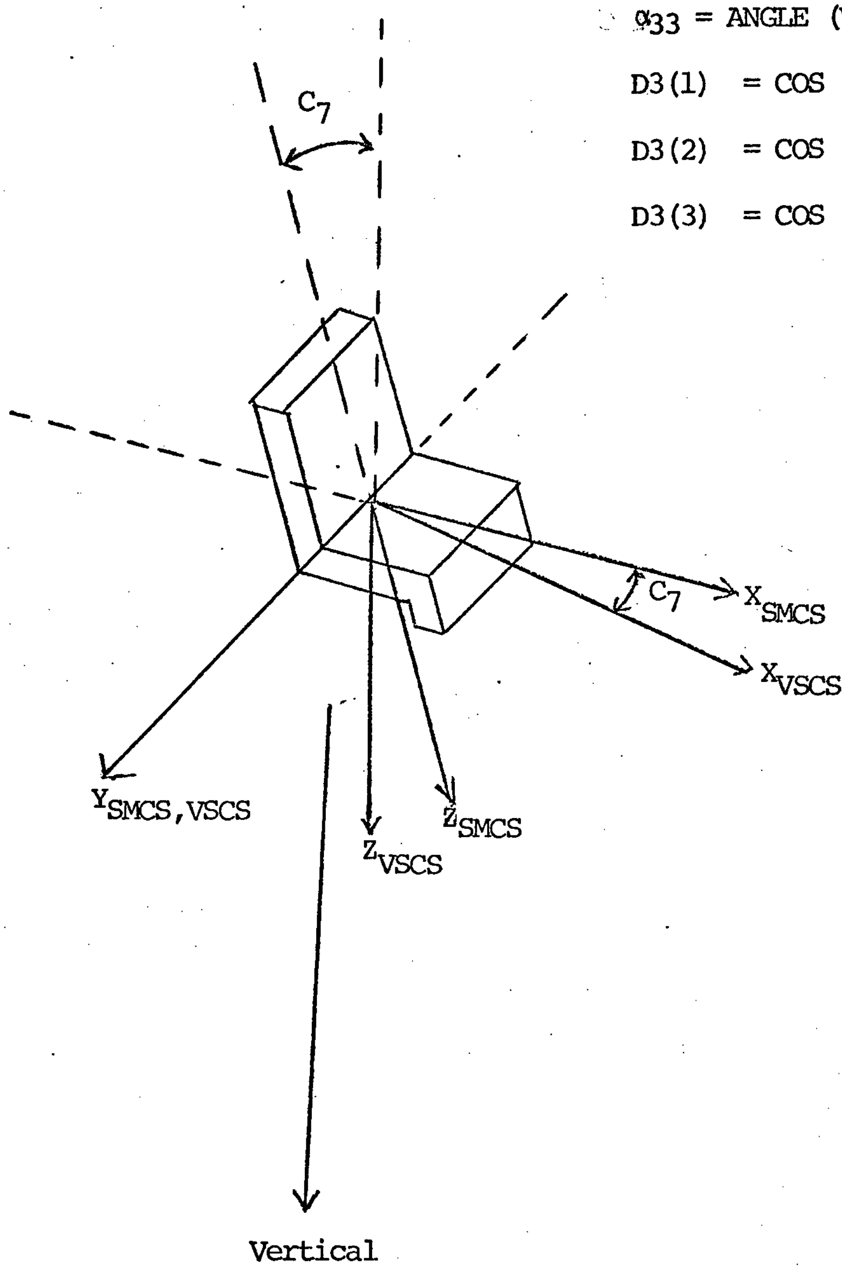

Figure 1 - VSCS, SMCS Definition

ASE is a direction cosine matrix which transfers a vector from SMCS to the EFCS. The initial values of the three direction cosine matrix elements, D3(1), D3(2), and D3(3) are calculated as follows:

$$D3(I) = -ASEM\ (I,2)/(\sum_{J=1}^{3} ASEM\ (I,J)^2)^{\frac{1}{2}}$$

Where I = 1,3.

2.3 Integrate Rotational Rate Data

The rotational rates are integrated with respect to time in order to keep track of the rotation of the seat/occupant system about each of the axes. The integration scheme is based on the following formula for the trapezoidal rule:

$$A = \tfrac{1}{2}\,\Delta h(Y_1 + 2Y_2 + 2Y_3 + \cdots + Y_n)$$

where Y_i corresponds to rotational rates for a particular time. Since data is sampled approximately every 1.25 ms, $\Delta h = 1.25$.

The program does not update the rotation position about each of the seat/occupant axes at every time increment but only when the rotation has exceeded a predefined angle. A is defined as the maximum angle that could be generated in one sampling period of rotating at the maximum rate that can be sensed by the gyroscopes. In this case, the maximum rate that can be sensed by the gyroscopes. In this case, the maximum rate is 8.72 rad/sec and the sampling period is 1.25 ms; therefore A = .0109 radians. If however, the seat/occupant system is rotating at the maximum rate in a negative direction (-8.72 rad/sec), then A = -.0109

radian. When the seat/occupant system has rotated at least .0109 rad. in either direction around any of the 3 axes, the appropriate direction cosines are updated and the corresponding test value is subtracted from the sum; the integration then continues as before.

2.4 Update Direction Cosines

Implementation of this part of the vertical seeking algorithm is based on the Crowder-Hession Direction Cosine Updating Algorithm as follows:

$$C_{31,a} = C_{31,k}$$

$$C_{32,a} = C_{32,k} - \Delta\Theta_Z\, C_{31,k}$$

$$C_{33,a} = C_{33,k} + \Delta\Theta_Y\, C_{31,k}$$

$$C_{31,b} = C_{31,a} + \Delta\Theta_Z\, C_{32,a}$$

$$C_{32,b} = C_{32,a}$$

$$C_{33,b} = C_{33,a} - \Delta\Theta_X\, C_{32,a}$$

$$C_{31,k+1} = C_{31,b} - \Delta\Theta_Y\, C_{33,b}$$

$$C_{32,k+1} = C_{32,b} + \Delta\Theta_X\, C_{33,b}$$

$$C_{33,k+1} = C_{33,b}$$

Where:

C_{31}, C_{32}, C_{33} are the three direction cosines D3(1), D3(2), D3(3)

K, K + 1 designate points in time;

a, b are values for intermediate calculations used in arriving at the next point in time and

$\Delta\Theta_X$, $\Delta\Theta_Y$, $\Delta\Theta_Z$ are the angular displacements of the gyros.

The displacements are represented by a quantum pulse weight $\Delta\Theta$ which has the general form of 2^{-n}, where n is determined by the sampling time and the maximum value of the angular rate is given as 7 (2^{-7} = .0078 rad).

Every 2.5 ms, this update algorithm is processed twice. In processing the algorithm, $\Delta\Theta_X$, $\Delta\Theta_Y$, and $\Delta\Theta_Z$ are determined by the integration of the rotational rates around the X, Y, and Z axes respectively. If the seat/occupant system has rotated less than .0109 radians in either the positive or the negative direction, the corresponding $\Delta\Theta j = 0.0$. If the seat/occupant system has rotated at least $\pm$ 0.0109 radians, the corresponding $\Delta\Theta j = \pm 2^{-7} = \pm 0.0078$ radian.

2.5 Generate Rocket Gimballing Commands

Two commands, a roll command PC and a pitch command QC, are output to gimbal the rocket and are governed by a set of rules as follows:

1. If the Z-Axis of the VSCS is parallel to the normal to the earth surface, a maximum roll command of 2.0 is generated;
2. If the seat/occupant is in an inverted position, (D3(3), < 0.0), a large command (between 1 and 2) is generated in order to rotate the seat/occupant back into the upright position quickly;
3. If the seat/occupant is in an upright attitude, a moderate pitch/roll command is generated according to the values of D3(1) and D3(2);
4. The extreme values for either PC or QC are limited to +2.0 and -2.0.

The rocket gimballing command control law is given in Figure 2.

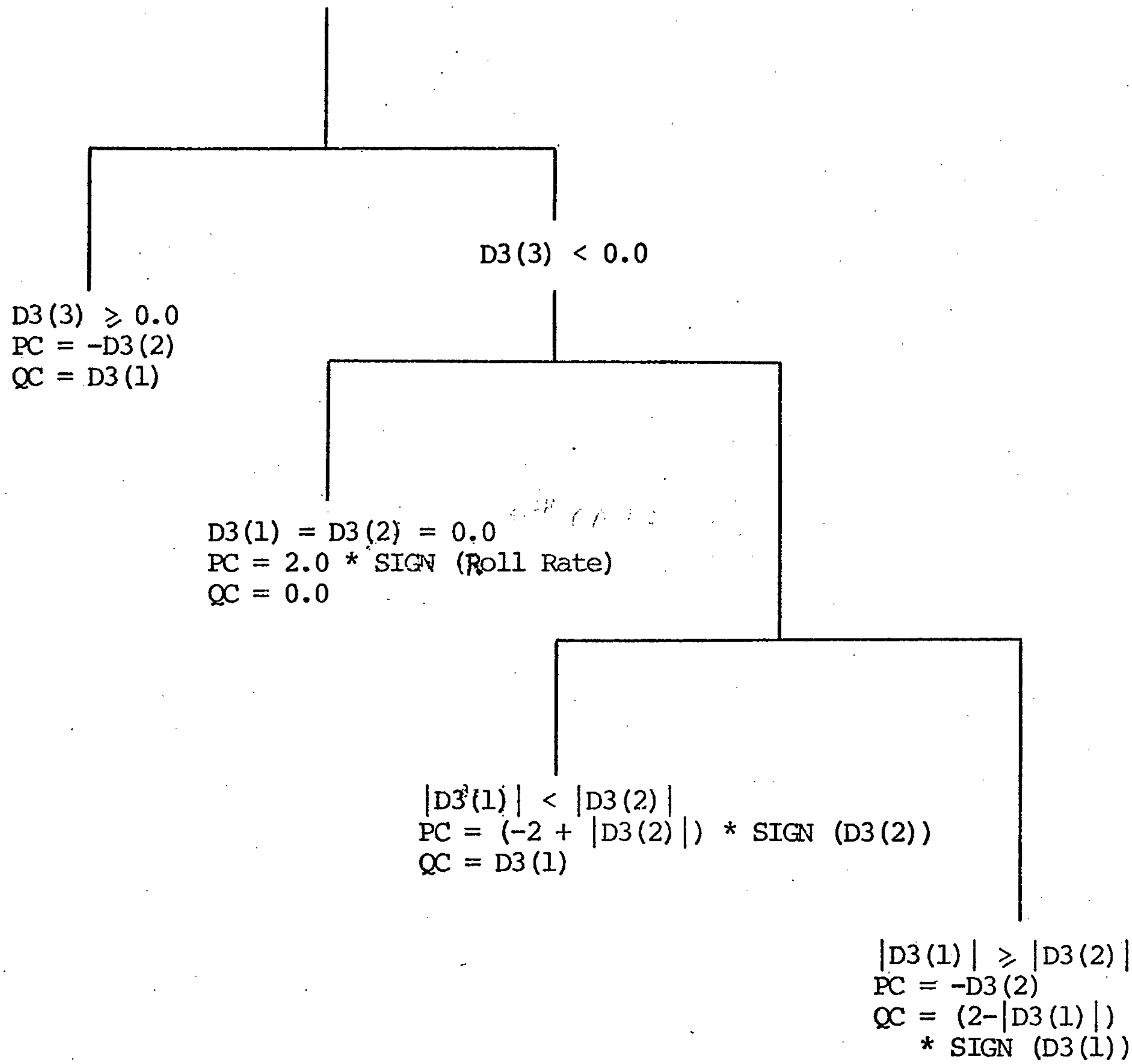

Figure 2 - Rocket Gimballing Command Control Law

2.6 Gimbal Rocket

The seat-rocket motor arrangement as implemented at NWC is shown in Figure 3. It can be seen from the figure that the rocket motor can be gimballed $\pm$ 16° in two directions.

The roll command, PC, and the pitch command, QC, are output to a Signal converter, which in turn causes the rocket to move along the X-Axis and/or Y-Axis of VSCS. The displacement of the angle is measured not from the previous position of the rocket but from the neutral position.

There are two sets of equations involved in simulating this signal conversion:

The roll thrust vector control angle, X14, is calculated as follows:

$$\phi_T = (C_{76} * PC) - (C_{77} * X_4)$$
$$\dot{X}_{16} = \phi_T - (100 * X_{16})$$
$$\phi_T = 5 * (\dot{X}_{16} + (20 * X_{16}))$$
$$\dot{X}_{35} = (31.45 * \phi_T) - (31.45 * X_{35})$$
$$\dot{X}_{36} = X_{35} - (90992 * X_{14}) - (48.26 * X_{36})$$
$$\dot{X}_{14} = X_{36}$$

X_{14} is obtained through integration over $\dot{X}_{14}$.

X_4 is the body roll rate and PC is the roll command.

The pitch thrust vector control angle, X_{13}, is calculated as follows:

SEAT MOTOR ARRANGEMENT

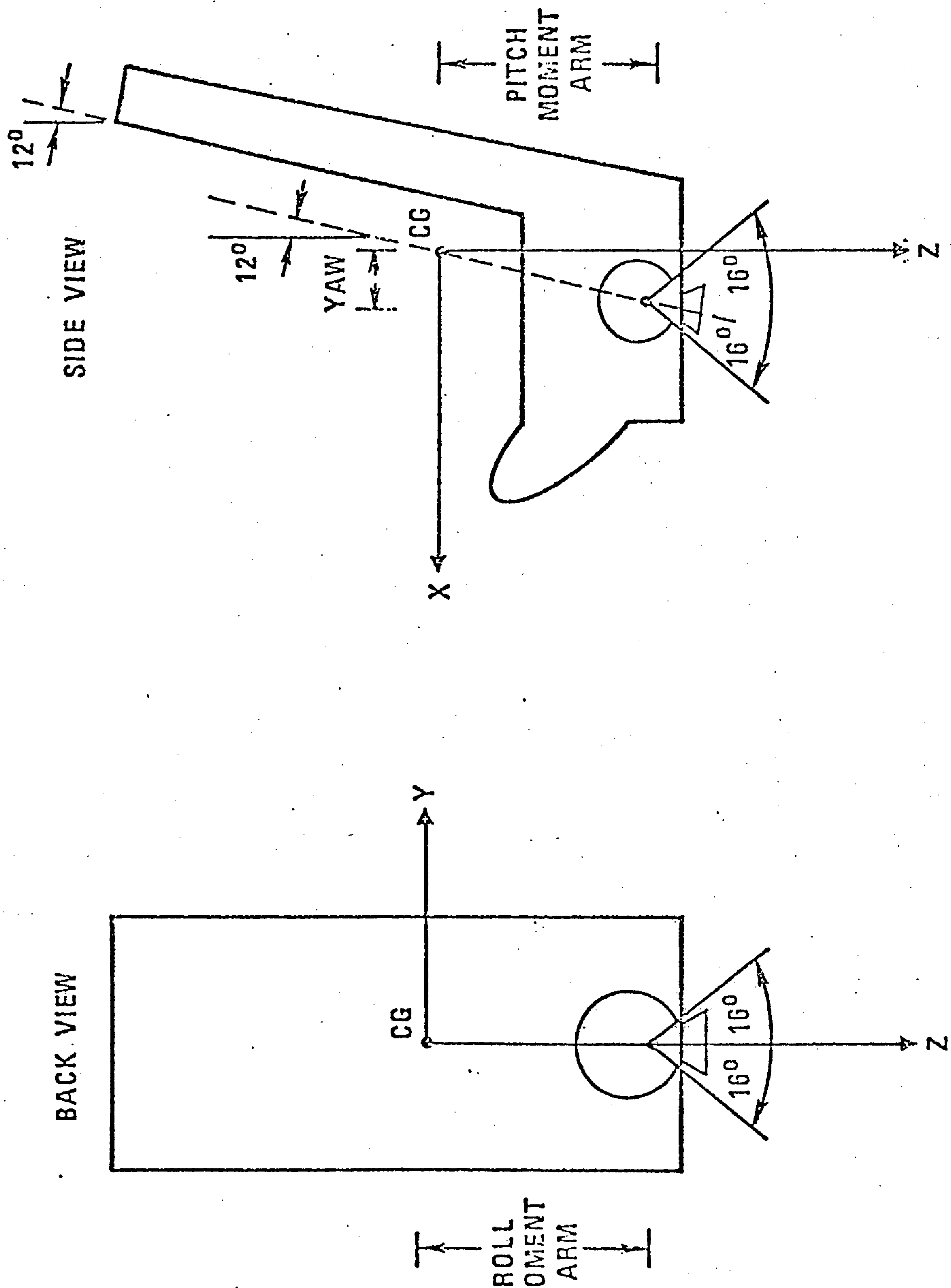

Figure 3 - Seat Motor Arrangement

$$\Theta_T = (C_{74} * QC) - (C_{75} * X_5)$$

$$\dot{X}_{15} = \Theta_T - (100 * X_{15})$$

$$\Theta_T = 5 * (\dot{X}_{15} + (20 * X_{15}))$$

$$\dot{X}_{40} = (31.45 * \Theta_T) - (31.45 * X_{40})$$

$$\dot{X}_{41} = X_{40} - (90992 * X_{13}) - (48.26 * X_{41})$$

$$\dot{X}_{13} = X_{41}$$

The pitch angle X_{13} is obtained through integration over $\dot{X}_{13}$. X_5 is the body pitch rate and QC is the pitch command.

3.0 PROGRAM DESCRIPTION

The following discussion describes the modifications that were made to the ICARUS program in order to include a vertical seeking maneuver simulation capability.

3.1 Program ICARUS

Two common blocks were added to program ICARUS in order to provide the inter-subroutine data communication related to the vertical seeking maneuver simulation. Common block MPESIN contains the input data and the related constants and common block MPES contains intermediate variables used in computations.

```
COMMON/MPESIN/PSM, QSM, RSM, IMPES, IMRKT, IMCAT, IMRE1, IMRE2, C5,
            C6, C7, C74, C75, C76, C77, DVC(3,3).
COMMON/MPES/D3(3), DALPH(3), DTH(3,2), ANGR(3), PC, QC, X13, X14, X15,
            X16, X35, X36, X40, X41, XD13, XD14, XD15, XD16, XD35, DX36,
            XD40, XD41, PTIME, IUPDATE, PPH, PTHAT.
```

3.2 Subroutine EXECUT

Flags IMPES, IMRKT and IMCAT are set according to appropriate input values.

IMPES - Vertical seeking seeking maneuver flag, set by INT(10)

= Ø ; OFF

= 1 ; ON, with non-filter version

= 2 ; ON, with filter version

IMRKT - Rocket Ignition time, set by INT(11).

= 1 ; time equal to TRI

= 2 ; calculated rocket ignition time

IMCAT - Catapult flag, set by INT(12).

= Ø ; no catapult

= 1 ; catapult

IMRE1 - Work array index before drogue chute projection.

IMRE2 - Work array index after drogue chute projection.

3.3 Subroutine INIT

A set of constants was defined through data statements.

C5 = .2792

C6 = .2792

C74 = .54

C75 = .09

C76 = .54

C77 = .09

Angle C7, which is the angle between the Z-Axis of the SMCS and the Z-Axis of the Vertical Seeking Coordinate System (VSCS), is defined in input. Matrix DVC transfers the data from the VSCS to the SMCS.

DVC(1,1) = COS (C7)

DVC(1,2) = 0.0

DVC (1,3) = SIN (C7)

DVC(2,1) = 0.0

DVC(2,2) = 1.0

DVC(2,3) = 0.0

DVC(3,1) = -SIN(C7)

DVC(3,2) = 0.0

DVC(3,3) = COS(C7)

Matrix ASE, which transfers the data from SMCS to EFCS, along with Matrix DVC, was used in the computation of the initial values D3(1), and D3(2) and D3(3) of the direction cosine matrix.

ASEM (1,1) = (DVC(1,1) * ASE (1,1)) + (DVC (1,2) * ASE (2,1)) +
(DVC (1,3) * ASE (3,1))

ASEM (1,2) = (DVC(1,1) * ASE (1,2)) + (DVC (1,2) * ASE (2,2)) +
(DVC (1,3) * ASE (3,2))

ASEM (1,3) = (DVC (1,1) * ASE (1,3))+ (DVC (1,2) * ASE (2,3)) +
(DVC (1,3) * ASE (3,3))

ASEM (2,1) = (DVC (2,1) * ASE (1,1)) + (DVC (2,2) * ASE (2,1)) +
(DVC (2,3) * ASE (3,1))

ASEM (2,2) = (DVC (2,1) * ASE (1,2)) + (DVC (2,2) * ASE (2,2)) +
(DVC (2,3) * ASE (3,2))

ASEM (2,3) =(DVC (2,1) * ASE (1,3))+(DVC (2,2) * ASE (2,3))+
(DVC (2,3) * ASE (3,3))

ASEM (3,1) =(DVC (3,1) * ASE (1,1))+(DVC (3,2) * ASE (2,1))+
(DVC (3,3) * ASE (3,1))

ASEM (3,2) =(DVC (3,1) * ASE (1,2))+(DVC (3,2) * ASE (2,2))+
(DVC (3,3) * ASE (3,2))

ASEM (3,3) =(DVC (3,1) * ASE (1,3))+(DVC (3,2) * ASE (2,3))+
(DVC (3,3) * ASE (3,3))

$$D3(1)=(-ASEM(1,2))/\sqrt{(ASEM(1,1))^2+(ASEM(1,2))^2+(ASEM(1,3))^2}$$

$$D3(2)=(-ASEM(2,2)/\sqrt{(ASEM(2,1))^2+(ASEM(2,2))^2+(ASEM(2,3))^2}$$

$$D3(3)=(-ASEM(3,2))/\sqrt{(ASEM(3,1))^2+(ASEM(3,2))^2+(ASEM(3,3))^2}$$

3.4 Subroutine INPUT

Angle C7, which is the angle between the Z-Axis of the SMCS and the Z-Axis of the vertical seeking coordinate system (VSCS), is defined by input INT(13). Rocket thrust tables TRR and TLR were converted from the standard burn time into the TRBO burn time for each individual test case. The rocket thrust was adjusted according to the difference in rocket burn time in order to maintain the same total momentum over the whole time period.

The total momentum from the standard rocket thrust table is calculated as

$$OLD = \sum_{I=1}^{NPTS-1} (YTAB(I,2) + YTAB((I+1),2)) * (YTAB((I+1),1) - YTAB(I,1))/2.0$$

Where YTAB (I,2) = Rocket thrust at point I

YTAB (I,1) = Rocket burn time at point I

NPTS = Total number of points

The time ratio between the TRBO and the standard rocket burn out time is R.

R = TRBO/(YTAB (NPTS, 1) - YTAB (1,1))

The converted rocket thrust and corresponding time at point I is calculated as:

TNEW (I,2) = YTAB (I,2)/R

TNEW (I,1) = TNEW ((I-1), 1) + (YTAB (I,1) - YTAB ((I-1), 1)) * R

The total momentum from the converted rocket thrust table is calculated as:

$$\text{NEW} = \sum_{I=2}^{\text{NPTS}} (\text{TNEW}((I-1),2) + \text{TNEW}(I,2)) * (\text{TNEW}(I,1) - \text{TNEW}((I-1),1))/2.0$$

If the total momentum NEW is beyond (1 $\pm$ .02) of the total momentum OLD, the array TNEW and NEW are recalculated with

R = R * NEW/OLD

Both the standard thrust table and the converted thrust table are printed for user's reference.

3.5 Subroutine ROCKFM

a) The angular increment for every time increment of 0.00125 second is computed as:

DTH (I, 2) = ANGR (I) * 0.00125

Where I = 1, 3

ANGR = Angular Rate

The total angular increment from the previous update is:

TDTH = DTH (I,2) + DTH (I,1)

Where

DTH (I,1) = Previous angular increment, (the value is always less than 0.0109 Rad.)

If TDTH is greater than 0.0109, set

$$DTH(I,2)=0.0078125*\left(\frac{TDTH}{|TDTH|}\right)$$

DTH (I,1) = TDTH - DTH (I,2)

Otherwise, set

DTH (I,2) = 0.0

DTH (I,1) = TDTH

b) Update the direction cosine matrix elements D3(1), D3(2) and D3(3) by the Crowder-Hession algorithm.

D3(2) = D3(2) - (DTH (3,2) * D3(1))

D3(3) = D3(3) + (DTH (2,2) * D3(1))

D3(1) = D3(1) + (DTH (3,2) * D3(2))

D3(3) = D3(3) - (DTH (1,2) * D3(2))

D3(1) = D3(1) - (DTH (2,2) * D3(3))

D3(2) = D3(2) + (DTH (1,2) * D3(3))

c) The commands PC and QC to move the rocket to a new position are decided based upon the value of D3(1), D3(2) and D3(3).

D3(3) > 0.0
PC =-D3(2)
QC = D3(1)

D3(3) > 0.0

D3(1) = D3(2) = 0.0
PC = 2.0 * SIGN (ROLL RATE)
QC = 0.0

|D3(1)| < |D3(2)|
PC = (-2 + 1 |D3(2)|) * SIGN (D3(2))
QC = D3(1)

|D3(1)| > |D3(2)|
PC = -D3(2)
QC = (2-|D3(1)|) * SIGN (D3(1))

d) The next pitch angle of the rocket is calculated as

THET = (C74 * QC) - (C75 * ANGR (2))

$\dot{X}15$ = THET - (100 * X15)

THET = 5 * ($\dot{}$15 + (20 * X15))

Restrict angle THET to within ± C5.

THET = MAX (THET, - C5)

THET = MIN (THET, C5)

$\dot{X}13$ is calculated as

$$\dot{X}40 = (31.45 * THET) - (31.45 * X40)$$

$$\dot{X}41 = X40 - (90992 * X13) - (48.26 * X41)$$

$$\dot{X}13 = X41$$

The pitch angle X13 is obtained through integration over $\dot{X}13$.

e) The next roll angle of the rocket is calculated as:

$$TPHI = (C76 * PC) - (C77 * ANGR(1))$$

$$\dot{X}16 = TPHI - (100 * X16)$$

$$TPHI = 5 * (\dot{X}16 + (20 * X16))$$

Restrict angle TPHI to within $\pm$ C5.

$$TPHI = MAX\ (TPHI, -C6)$$

$$TPHI = MIN\ (TPHI, C6)$$

The roll angle X14 is obtained through integration over $\dot{X}14$ which is calculated as:

$$\dot{X}35 = (31.45 * TPHI) - (31.45 * X35.)$$

$$\dot{X}36 = X35 - (90992 * X14) - (48.26 * X36.)$$

$$\dot{X}14 = X36.$$

3.6 <u>Subroutine SEATMAN</u>

The call to subroutine ROCKFM in subroutine SEATMAN has been changed from CALL ROCKFM (QT) to CALL ROCKFM (QT, QY, QDYDT).

4.0 Summary

CSC completed the modifications to the ICARUS program described in Section 3 and began preliminary testing.